PRG

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/598,713	06/21/2000	Jonathan P. Douglas	2207/8613	7081

23838 7590 06/20/2003

KENYON & KENYON
1500 K STREET, N.W., SUITE 700
WASHINGTON, DC 20005

EXAMINER
HUISMAN, DAVID J

ART UNIT	PAPER NUMBER
2183	2

DATE MAILED: 06/20/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/598,713	DOUGLAS, JONATHAN P.
	Examiner	Art Unit
	David J. Huisman	2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 June 2000* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☒ Claim(s) *18 and 19* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *21 June 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. Claims 1-20 have been examined.

Specification

2. The abstract of the disclosure is objected to because it does not accurately describe the contents of the disclosure. No mention is made of how the system deals with call and return instructions, yet, the disclosure deals mostly with these concepts. Correction is required. See MPEP § 608.01(b).

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Drawings

4. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference sign(s) not mentioned in the description: In Fig.3, reference number 500 has not been mentioned within the disclosure. A proposed drawing correction, corrected drawings, or amendment to the specification to add the reference sign(s) in the description, are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

5. The drawings are objected to because of the following minor informality: It is difficult to see reference number 540 in Fig.3, due to its placement within the hatched component. A proposed drawing correction or corrected drawings are required in reply to the Office action to

avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Objections

6. Claim 18 is objected to because of the following informalities: The examiner recommends inserting --of cascaded pipestages-- after "the second plurality" in line 15. Also, there is lack of antecedent basis for "the one pipestage" in line 17, since claim 17 refers to both a first and second pipeline stages. Appropriate correction is required.

7. Claim 19 is objected to because of the following informalities: The examiner recommends inserting --of cascaded pipestages-- after "the first plurality" in line 21. Also, there is lack of antecedent basis for "the one pipestage" in line 23, since claim 17 refers to both a first and second pipeline stages. Appropriate correction is required.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 1, 4, 6-7, and 9-16 are rejected under 35 U.S.C. 102(b) as being anticipated by Hennessy & Patterson, Computer Architecture - A Quantitative Approach, 2nd Edition, 1996 (herein referred to as Hennessy).

10. Referring to claim 1, Hennessy has taught an instruction pipe control method comprising:

a) reading a new instruction from an instruction pipestage. See page 154, Figure 3.13 and note that each instruction is read from the appropriate stage. For instance, in clock cycle 1 (CC1), the LW instruction is read from the fetch stage, in CC2, the LW instruction is read from the decode stage, and so on.

b) determining, with reference to other instructions read previously from the instruction pipestage, whether valid data associated with the new instruction can be written to a next instruction pipestage. See page 154, Figure 3.13 and note, for instance, that if the SUB instruction were to be read from the execution stage in CC4, a hazard would exist between the SUB instruction and the previous LW instruction (since the SUB instruction is dependent on data produced by the LW instruction). If no hazard were to exist, then the SUB instruction could be executed in CC4 and valid data associated with the instruction could be written to a next pipeline stage.

c) stalling processing of the new instruction until valid data associated with the new instruction can be written to the next instruction pipestage. Since a hazard exists between the LW instruction and the SUB instruction, the SUB's associated data cannot be written to the next pipestage. As a result, the system inserts a bubble, or stalls for a cycle. Again, see Figure 3.13 on page 154.

11. Referring to claim 4, Hennessy has taught an instruction pipe control method as described in claim 1. Hennessy has further taught that if the new instruction is a call instruction, the determining includes determining whether immediate processing of the call instruction would exceed a predetermined access rate associated with a shared resource. Note that Hennessy has taught the basic concept of a call instruction on page 277. It should be further noted that each

instruction within the instruction set travels through the processing pipeline, as shown on page 142, Figure 3.6. In this particular figure, if Instruction 3 were a call instruction, the immediate processing of the call would cause a structural hazard with the load instruction since both are accessing a memory, which is shared by the fetch and memory stages of the pipeline. In this case, the immediate processing of the call would exceed a predetermined access rate associated with the memory, where the access rate is one instruction per cycle.

12. Referring to claim 6, Hennessy has taught an instruction pipe control method as described in claim 1. Hennessy has further taught that the stalling stalls the instruction pipestage and all other instruction pipestages before it in the instruction pipe. See Figure 3.13 on page 154 and note that if the SUB instruction is stalled before its 3rd stage (as shown), then the next two subsequent instructions are stalled before their 2nd and 1st stages respectively.

13. Referring to claim 7, Hennessy has taught an interface method for an instruction pipe that shares access to an external resource, comprising:

a) reading a new instruction from an instruction pipestage. See page 142, Figure 3.6 and note that each instruction is read from the appropriate stage. For instance, in clock cycle 1 (CC1), the LW instruction is read from the fetch stage, in CC2, the LW instruction is read from the decode stage, and so on.

b) if the new instruction requires access to the external resource, determining with reference to other instructions read previously from the instruction pipestage, whether immediate processing of the new instruction would cause the instruction pipe to exceed an access allocation for the instruction pipe. Again, see page 142, Figure 3.6. In this particular figure, if Instruction 3 were a call instruction, the immediate processing of the call would cause a structural hazard with the

load instruction since both are accessing a memory, which is shared by the fetch and memory stages of the pipeline. In this case, the immediate processing of the call would exceed a predetermined access rate associated with the memory, where the access rate is one instruction per cycle.

c) if so, stalling the new instruction. See page 143, Figure 3.7. Note that the structural hazard is dealt with by stalling.

14. Referring to claim 9, Hennessy has taught a method as described in claim 7. Hennessy has further taught that the stalling stalls the instruction pipestage and all other instruction pipestages before it in the instruction pipe. See Figure 3.13 on page 154 and note that if the SUB instruction is stalled before its 3rd stage (as shown), then the next two subsequent instructions are stalled before their 2nd and 1st stages respectively.

15. Referring to claim 10, Hennessy has taught a method for interfacing an instruction pipe with an external resource characterized by a predetermined round-trip communication latency period, the method comprising:

a) reading a new instruction from an instruction pipe stage. See page 154, Figure 3.13 and note that each instruction is read from the appropriate stage. For instance, in clock cycle 1 (CC1), the LW instruction is read from the fetch stage, in CC2, the LW instruction is read from the decode stage, and so on.

b) determining, with reference to other instructions read previously from the instruction pipestage, whether valid data associated with the new instruction is available to the instruction pipe prior to expiration of the round-trip communication latency period. It should be realized that this constitutes basic dependency checking. For instance, on page 154, in Figure 3.12, a

RAW hazard exists between the LW instruction and the SUB instruction (i.e., the SUB instruction needs to read an operand that is produced by the LW instruction). Therefore, since the SUB is dependent on the LW, it will be determined that the data associated with the SUB is not available prior to the expiration of the round-trip communication latency (that is, the latency required for the load to access memory). On the other hand, if a SUB instruction immediately following the LW is not dependent on the LW (and causes no other hazards), then it would be determined that the data for that instruction would be available prior to the load finishing its access of memory.

c) if not, stalling processing of the new instruction until the round-trip communication latency period expires. If the data were not available, in the case of the RAW hazard, then the SUB instruction would have to be stalled until the LW instruction finishes accessing memory. See Figure 3.13 on page 154.

16. Referring to claim 11, Hennessy has taught a method as described in claim 10. Hennessy has further taught determining whether the new instruction requires access to the external resource in excess of an access allocation for the instruction pipe, and if so, stalling the new instruction. It should be noted that each instruction within the instruction set travels through the processing pipeline, as shown on page 142, Figure 3.6. In this particular figure, if Instruction 3 were the new instruction, the immediate processing of the new instruction would cause a structural hazard with the load instruction since both are accessing a memory, which is shared by the fetch and memory stages of the pipeline. In this case, the immediate processing of Instruction 3 would exceed an access allocation for the instruction pipe, where the access allocation for the pipeline is one instruction per cycle is allowed to access the memory.

17. Referring to claim 12, Hennessy has taught a method as described in claim 10. Hennessy has further taught that the stalling stalls the instruction pipestage and all other instruction pipestages before it in the instruction pipe. See Figure 3.13 on page 154 and note that if the SUB instruction is stalled before its 3rd stage (as shown), then the next two subsequent instructions are stalled before their 2nd and 1st stages respectively.

18. Referring to claim 13, Hennessy has taught in an instruction pipe, a clock throttling mechanism provided between a pair of instruction pipestages, comprising:

a) a state machine coupled to an output of a first instruction pipestage.

b) a clock control circuit having an input for a system clock signal and having an output for a modified clock signal, the output coupled to the pair of instruction pipestages, the clock control circuit controlled by the state machine.

See page 134 and 154. It should be noted that instructions progress to a next stage within the pipeline according to a clock pulse. If a stall is required at a particular stage, then a clock pulse is not applied to that stage, thereby not allowing the contents of that stage to progress to the next stage. Therefore, in order to determine whether a clock pulse should be applied or not, it must first be determined whether a hazard exists. In order to do this, the system would inherently include a state machine which would recognize the state of the machine (hazard exists or doesn't exist), and then supply the correct clock signals accordingly, in order to eliminate the hazard(s). Therefore, the clock control would inherently be connected to the state machine such that when a hazard is detected, the appropriate clock value would be sent to each pipeline stage.

19. Referring to claim 14, Hennessy has taught a mechanism as described in claim 13. Hennessy has further taught a read/write controller under control of the state machine and having

an output for controlling writes to the second instruction pipestage. See Figure 3.4 on page 134. Any component which reads from one of the pipe registers and writes to a pipe register can be considered a read/write controller. For instance, the ALU can be considered a read/write controller in that it reads operands from the ID/EX register and writes a result to the EX/MEM register. However, if a stall is needed in the EX stage due to some type of hazard, then a clock pulse will not be provided by the state machine to that stage. By not clocking the ALU, it will not be able to read/write results from/to pipeline registers.

20. Referring to claim 15, Hennessy has taught a mechanism as described in claim 13. Hennessy has further taught:

a) a first register coupled to the first instruction pipestage. See Figure 3.4 on page 134 and Figure 3.5 on page 136, and note the ID/EX.A register (portion of ID/EX that is sent to the lower input of the top MUX which is fed into the ALU). This register represents a result obtained during the decode stage.

b) a second register. See Figure 3.4 on page 134 and Figure 3.5 on page 136, and note the ID/EX.B register (portion of ID/EX that is sent to the upper input of the bottom MUX which is fed into the ALU).

c) a selector coupled to the first and second registers and having an output coupled to the second instruction pipestage. Note the MUXs that are connected to the register outputs. The MUXs then have outputs which are ultimately fed into the EX/MEM register, which holds results obtained during the EX stage.

21. Referring to claim 16, Hennessy has taught a mechanism as described in claim 15. Hennessy has further taught that the selector is controlled by the state machine. As described

above, the state machine can be ion two general states (one being where hazards are detected and a stall is needed and the other where no hazards exist and all instructions continue to progress through the pipeline). Therefore, the state machine will control whether the MUX is inputting and outputting data which will be operated on and whose destination is a next pipeline stage, or inputting and outputting data which will not be operated on and sent to a next pipeline stage.

Claim Rejections - 35 USC § 103

22. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

23. Claims 2, 5, and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hennessy, as applied above, in view of Hoyt et al., U.S. Patent No. 5,604,877 (herein referred to as Hoyt).

24. Referring to claim 2, Hennessy has taught an instruction pipe control method as described in claim 1. Hennessy has not explicitly taught that if the new instruction is a return instruction, the determining includes determining whether a return address is available within the instruction pipe. However, Hoyt has taught a system for predicting return addresses wherein the prediction is made using either a register within the pipeline (Fig.5, component 45), or using a return stack buffer (Fig.5, component 51), which is normally implemented as a LIFO in main memory (column 2, lines 20-36). Hoyt has disclosed (in the Background section) that performing such prediction improves the efficiency of the system in that the processor can continue fetching

instructions down the predicted path, thereby keeping the pipeline full. This advantage is also recognized by Hennessy (as shown in Figure 3.26 on page 167). Therefore, in order to aid in the prediction of return addresses, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hennessy such that it is capable of predicting such addresses, as taught by Hoyt. And, part of this prediction system includes determining whether a return address is available within the instruction pipe (from Fig.5, component 45). This register is used to try to speed up the prediction in that by using the register, a more expensive access to the return stack buffer would be unnecessary.

25. Referring to claim 5, Hennessy has taught an instruction pipe control method as described in claim 4. Hennessy has not explicitly taught after the stalling terminates, storing a return address associated with the call instruction both locally and in a shared resource. However, Hoyt has taught a system for predicting return addresses wherein the prediction is made using either a register within the pipeline (Fig.5, component 45), or using a return stack buffer (Fig.5, component 51), which is normally implemented as a LIFO in main memory (column 2, lines 20-36). Hoyt has disclosed (in the Background section) that performing such prediction improves the efficiency of the system in that the processor can continue fetching instructions down the predicted path, thereby keeping the pipeline full. This advantage is also recognized by Hennessy (as shown in Figure 3.26 on page 167). Therefore, in order to aid in the prediction of return addresses, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hennessy such that it is capable of predicting such addresses, as taught by Hoyt. And, part of this prediction system includes storing a return address both locally (in register 45, Fig.5) and in a shared resource (stack 51, Fig.5), where the return stack buffer is

shared by components 40 and 60 (decoder and branch target buffer circuit). By storing the prediction locally, the prediction could be accessed more quickly as opposed to retrieving the prediction from the return stack buffer in memory. However, the return addresses must also be stored in the stack since the register can only hold one prediction. If there were no stack, then the system would lose return predictions if two or more calls happen in a row before a return. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to store the address both locally and in a shared resource.

26. Referring to claim 8, Hennessy has taught a method as described in claim 7. Hennessy has taught a call instruction (page 277) but has not taught that if the new instruction is a.call instruction, after the stalling terminates, storing a return address associated with the call instruction both locally and in a shared resource. However, Hoyt has taught a system for predicting return addresses wherein the prediction is made using either a register within the pipeline (Fig.5, component 45), or using a return stack buffer (Fig.5, component 51), which is normally implemented as a LIFO in main memory (column 2, lines 20-36). Hoyt has disclosed (in the Background section) that performing such prediction improves the efficiency of the system in that the processor can continue fetching instructions down the predicted path, thereby keeping the pipeline full. This advantage is also recognized by Hennessy (as shown in Figure 3.26 on page 167). Therefore, in order to aid in the prediction of return addresses, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hennessy such that it is capable of predicting such addresses, as taught by Hoyt. And, part of this prediction system includes storing a return address both locally (in register 45, Fig.5) and in a shared resource (stack 51, Fig.5), where the return stack buffer is shared by components 40 and

60 (decoder and branch target buffer circuit). By storing the prediction locally, the prediction could be accessed more quickly as opposed to retrieving the prediction from the return stack buffer in memory. However, the return addresses must also be stored in the stack since the register can only hold one prediction. If there were no stack, then the system would lose return predictions if two or more calls happen in a row before a return. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to store the address both locally and in a shared resource.

27. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Skadron et al., Improving Prediction for Procedure Returns with Return-Address-Stack Repair Mechanisms, 1998 (herein referred to as Skadron).

28. Referring to claim 3, Hennessy has taught an instruction pipe control method as described in claim 1. Hennessy has not taught that if the new instruction is a return instruction, the determining includes determining whether sufficient time has expired from an earlier return instruction for a return address to be received from an external resource. Recall that Hennessy has taught the advantages of performing branch prediction. In essence, stalls can be reduced in the pipeline by continuing to fetch instructions along a predicted path. An example of this can be seen in Figure 3.26 on page 167. Skadron has further taught a system which focuses on return address prediction (for return instructions, which are also known as indirect branches). A person of ordinary skill in the art would have recognized that by modifying the system of Hennessy to include the teachings of Skadron, Hennessy would be able to predict return addresses as well, thereby keeping the pipeline full when a return instruction is encountered. Skadron has further

shown concern for two return instructions that occur in close succession (i.e., sufficient time has not expired from an earlier return before a second return is encountered). See page 261, column 2, first full paragraph. Skadron has further disclosed that this is a problem because if the first return cannot pop the stack before the second one looks in the stack, then the second one reads from the wrong location. A person of ordinary skill in the art would have recognized that in order to refrain from retrieving wrong predictions, each return instruction should wait for a previous return instruction to finish. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to determine whether sufficient time has expired from an earlier return instruction for a return address to be received from an external resource, in order to ensure that the correct locations in the external resource are accessed.

29. Claims 17-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Skadron, as applied above, in view of Hennessy, as applied above.

30. Referring to claim 17, Skadron has taught execution logic for a processor comprising:

a) a first instruction pipe, comprising a first plurality of cascaded pipestages. See Figure 4 on page 269 and note that at a branch, two paths are executed. Therefore, a first instruction pipe would be used to execute the predicted path for instance.

b) a return stack buffer provided in communication with at least one of the first pipestages. See Figure 4 on page 269 (see caption) and note that the two instruction pipes (paths) share a unified return-address stack.

c) a second instruction pipe, comprising:

c1) a second plurality of cascaded pipestages, at least one of the second pipestages provided in communication with the return stack buffer. See Figure 4 on page 269 and note the path referred to as the correct path (non-predicted path in this case), which is the second instruction pipe (the first being the predicted path, as described above). Note that both pipes (paths) would share the address stack.

c2) Skadron has not explicitly taught clock throttling logic coupled to the at least one second pipestage. However, as described above, Hennessy has shown on pages 134 and 154 that instructions progress to a next stage within the pipeline according to a clock pulse. If a stall is required at a particular stage, then a clock pulse is not applied to that stage, thereby not allowing the contents of that stage to progress to the next stage. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement clock-throttling logic in the second pipeline in order to handle hazards. By adjusting the clock (either supplying or not supplying a clock signal), stalls can be achieved and dependencies can be fixed.

31. Referring to claim 18, Skadron in view of Hennessy has taught execution logic as described in claim 17. Skadron has not explicitly taught:

a) a state machine coupled to an output of the one pipestage from the second plurality.

b) a clock control circuit having an input for a system clock signal and having an output for a modified clock signal, the output coupled to the one pipestage, the clock control circuit controlled by the state machine.

However, see page 134 and 154 of Hennessy. It should be noted that instructions progress to a next stage within the pipeline according to a clock pulse. If a stall is required at a particular stage, then a clock pulse is not applied to that stage, thereby not allowing the contents of that

stage to progress to the next stage. Therefore, in order to determine whether a clock pulse should be applied or not, it must first be determined whether a hazard exists. In order to do this, the system would inherently include a state machine which would recognize the state of the machine (hazard exists or doesn't exist), and then supply the correct clock signals accordingly, in order to eliminate the hazard(s). Therefore, the clock control would inherently be connected to the state machine such that when a hazard is detected, the appropriate clock value would be sent to each pipeline stage.

32. Referring to claim 19, Skadron in view of Hennessy has taught execution logic as described in claim 17. Skadron has not explicitly taught:

a) a state machine coupled to an output of the one pipestage from the first plurality.

b) a clock control circuit having an input for a system clock signal and having an output for a modified clock signal, the output coupled to the one pipestage, the clock control circuit controlled by the state machine.

However, see page 134 and 154 of Hennessy. It should be noted that instructions progress to a next stage within the pipeline according to a clock pulse. If a stall is required at a particular stage, then a clock pulse is not applied to that stage, thereby not allowing the contents of that stage to progress to the next stage. Therefore, in order to determine whether a clock pulse should be applied or not, it must first be determined whether a hazard exists. In order to do this, the system would inherently include a state machine which would recognize the state of the machine (hazard exists or doesn't exist), and then supply the correct clock signals accordingly, in order to eliminate the hazard(s). Therefore, the clock control would inherently be connected to the state

machine such that when a hazard is detected, the appropriate clock value would be sent to each pipeline stage.

33. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Skadron in view of Hennessy, as applied above, and further in view of Hoyt, as applied above.

34. Referring to claim 20, Skadron in view of Hennessy has taught execution logic as described in claim 17. Skadron in view of Hennessy has not explicitly taught that additional instruction pipestages from either the first or the second instruction pipe are provided in communication with the return stack buffer, the additional instruction pipestages also provided with additional clock throttling logic. However, Hoyt has taught a 4-stage return prediction pipeline in which multiple stages access the buffer. See column 10, lines 55-67, and column 13, lines 34-63, and note that both the fetch stage and decode stage result in accesses to the return address buffer. As a result, in a similar fashion to Figure 3.6 of Hennessy, a structural hazard could possible exist if two instructions result in an access of the buffer. Therefore, as described above, Hennessy has shown on pages 134 and 154 that instructions progress to a next stage within the pipeline according to a clock pulse. If a stall is required at a particular stage, then a clock pulse is not applied to that stage, thereby not allowing the contents of that stage to progress to the next stage. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement clock-throttling logic in each additional pipeline stage (that accesses the return address buffer) in order to handle hazards. By adjusting the clock (either supplying or not supplying a clock signal), stalls can be achieved and dependencies can be fixed.

Conclusion

35. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

D'Sa et al., U.S. Patent No. 6,151,671, has taught a system and method of maintaining and utilizing multiple return stack buffers.

Mehta, U.S. Patent No. 5,222,220, has taught microprocessor stack built-in guards.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

DJH
David J. Huisman
June 13, 2003

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100